INTRODUCCIÓN A LA FORENSIA DIGITAL

EXPLICACIÓN PRÁCTICA 1

TEMAS

- Encoding
- Magic numbers / File signatures
- Criptografía
- Hashing
- Esteganografía

CODIFICACIÓN (ENCODING)

- La codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural (como el de un alfabeto o silabario) en un símbolo de otro sistema de representación, aplicando **reglas de codificación**. Se trata de diferentes formas de representar un mismo contenido
- Para decodificar el mensaje solo hace falta conocer la regla, método o técnica empleada para la codificación del mismo (el algorítmo).
- Definen la forma en la que un caracter es codificado en un símbolo de otro sistema de representación.

EJEMPLOS DE SISTEMAS DE CODIFICACIÓN

- ASCII / Representación numérica en otras bases
- Morse
- Braile
- Lenguaje de Señas
- EBCDIC
- Base64

ASCII / BASE 2, 10 Y 16

Binario	Dec	Hex	Representación
0010 0000	32	20	espacio ( )
0010 0001	33	21	!
0010 0010	34	22	"
0010 0011	35	23	#
0010 0100	36	24	$
0010 0101	37	25	%
0010 0110	38	26	&
0010 0111	39	27	'
0010 1000	40	28	(
0010 1001	41	29	)
0010 1010	42	2A	*
0010 1011	43	2B	+
0010 1100	44	2C	,
0010 1101	45	2D	-
0010 1110	46	2E	.
0010 1111	47	2F	/
0011 0000	48	30	0
0011 0001	49	31	1
0011 0010	50	32	2
0011 0011	51	33	3
0011 0100	52	34	4
0011 0101	53	35	5

Binario	Dec	Hex	Representación
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U

Binario	Dec	Hex	Representación
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h
0110 1001	105	69	i
0110 1010	106	6A	j
0110 1011	107	6B	k
0110 1100	108	6C	l
0110 1101	109	6D	m
0110 1110	110	6E	n
0110 1111	111	6F	o
0111 0000	112	70	p
0111 0001	113	71	q
0111 0010	114	72	r
0111 0011	115	73	s
0111 0100	116	74	t
0111 0101	117	75	u

UNICODE

- Unicode es un estándar de codificación de caracteres diseñado para facilitar la transmisión y visualización de textos en numerosos idiomas.
- El término Unicode proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad.
- La versión 15.0 de Unicode contiene un repertorio de 149.186 caracteres

UNICODE

- Unicode puede ser implementado por diferentes codificaciones de caracteres como UTF-8, UTF-16, y UTF-32.

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

EBCDIC

- Código de Intercambio Decimal Codificado Binario Extendido

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX		PT			GE				FF	CR		
1	DLE	SBA	EUA	1C		NL				EM			DUP	SF	FM	ITB
2							ETB	ESC						ENQ		
3			SYN					EOT					RA	NAK		
4	SP										¢	.	<	(	+	
5	&										!	$	*	)	;	¬
6	-	/									|	,	%	_	>	?
7											:	#	@	'	=	"
8		a	B	c	d	e	f	g	h	I						
9		j	K	l	m	n	o	p	q	r						
A		~	S	t	u	v	w	x	y	z						
B																
C	{	A	B	C	D	E	F	G	H	I						
D	}	J	K	L	M	N	O	P	Q	R						
E	\		S	T	U	V	W	X	Y	Z						
F	0	1	2	3	4	5	6	7	8	9						

BASE64

Source character	V								m								0							
ASCII number	86								109								48							
Bit pattern	0	1	0	1	0	1	1	0	0	1	1	0	1	1	0	1	0	0	1	1	0	0	0	0
Base64 number	21								38								52							
Base64 character	V								m								0							

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

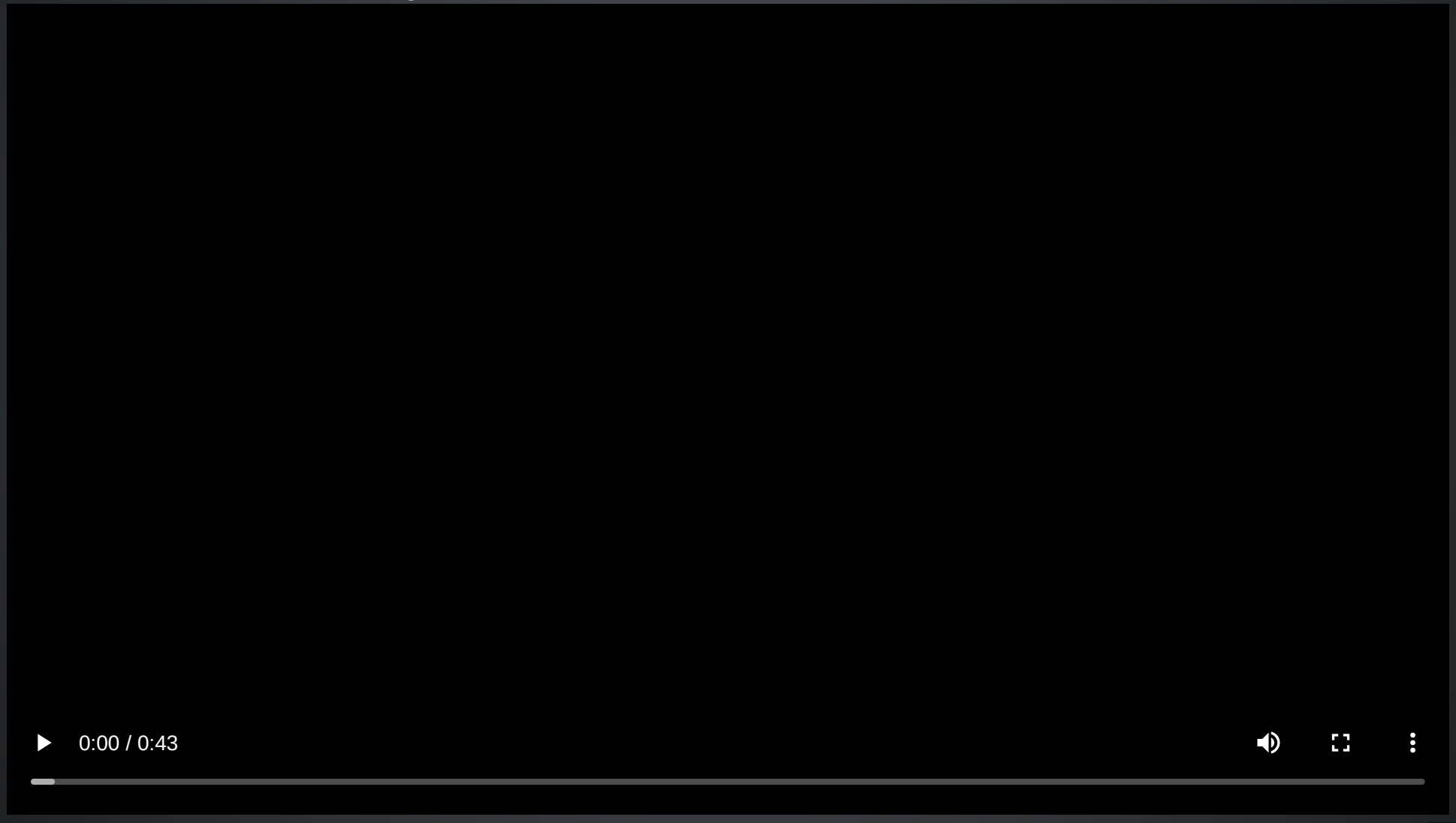
CODIFICACIÓN (ENCODING)

Ejemplo "HOLA CLASE"

```
Base64: "SE9MQSBDTEFTRQ=="

Hexa: "\x48\x4f\x4c\x41\x20\x43\x4c\x41\x53\x45"

Binario: "01001000 01001111 01001100 01000001 00100000 01000011 01001100 010

URL: "HOLA%20CLASE"

Morse: ".... --- .-.. .- ..... -.-. .-.. .- ... ."

Unicode: "U+0048 U+004F U+004C U+0041 U+0020 U+004 U+004C U+0041 U+0053 U+00
```

CODIFICACIÓN (ENCODING)

- Ejemplo "HOLA CLASE" en MORSE



CODIFICACIÓN (ENCODING)

```
$cat plane**.jpg**

▯▯▯▯
^L^L
◄◄►◄
^L↯▯↯◄▯◄◄◄▯▯
▯▯
⊛♈⌐♀♂%'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯
▯▯n▯g▯-
i▯▯▯▯◁▯z▯▯
C:⊛▯▯◁▯▯▯▯x=▯$o▯$Ÿ ▯▯◷A▷#♌▯&▯▯u▯▯`▯▯q▯▯5▯18◷▯▯y▯▯▯lL3▯8▯@▯B▯8▯  q▯l▯⌀JW▯▯▯
F ▯▯▯▯▯◷▯&8▯▯▯z▯◷▯▯5gcT▯▯▯▯◷p+3▯ZW▯▯▯▯▯▯▯▯I
```

MAGIC NUMBERS / FILE SIGNATURES (TIPOS DE ARCHIVOS)

- ¿Cómo saben que un archivo jpg es verdaderamente una imagen?
- Los **archivos** y **protocolos de comunicación** suelen comenzar con un valor determinado en sus primeros bytes para que los programas puedan saber de qué se trata. Incluso pueden tener más de un magic number válido para el mismo formato de archivo.
- **IMPORTANTE**: No confundir con "encabezados de archivos". Estos últimos no solo pueden contener un número mágico sino también el resto de los metadatos asociados al archivo.

MAGIC NUMBERS / FILE SIGNATURES (TIPOS DE ARCHIVOS)

$ cat plane.jpg | xxd

00000000: ffd8 ffe0 0010 **4a46 4946** 0001 0100 0001 .....**JFIF**.....
00000010: 0001 0000 ffdb 0043 0003 0202 0202 0203 .....C.....
00000020: 0202 0203 0303 0304 0604 0404 0404 0806 .....
00000030: 0605 0609 080a 0a09 0809 090a 0c0f 0c0a .....
00000040: 0b0e 0b09 090d 110d 0e0f 1010 1110 0a0c .....
00000050: 1213 1210 130f 1010 10ff db00 4301 0303 .....C...
00000060: 0304 0304 0804 0408 100b 090b 1010 1010 .....
00000070: 1010 1010 1010 1010 1010 1010 1010 1010 .....
00000080: 1010 1010 1010 1010 1010 1010 1010 1010 .....

MAGIC NUMBERS / FILE SIGNATURES (TIPOS DE ARCHIVOS)

En archivos:

- JPEG: "FF D8 FF E0 00 10 4A 46 49 46 00 01" / ... "JFIF"
- ZIP: "50 4B 03 04 / 50 4B 05 06" (empty archive) / ... "PK"
  https://en.wikipedia.org/wiki/List_of_file_signatures

En protocolos:

- SMB: FF 53 4D 42 "\xFF SMB"
- HTTP/2: 0x505249202a20485454502f322e300d0a0d0a534d0d0a
  "PRI * HTTP/2.0\r\n\r\nSM\r\n\r\n"
  https://en.wikipedia.org/wiki/Magic_number_(programming)#In_pro

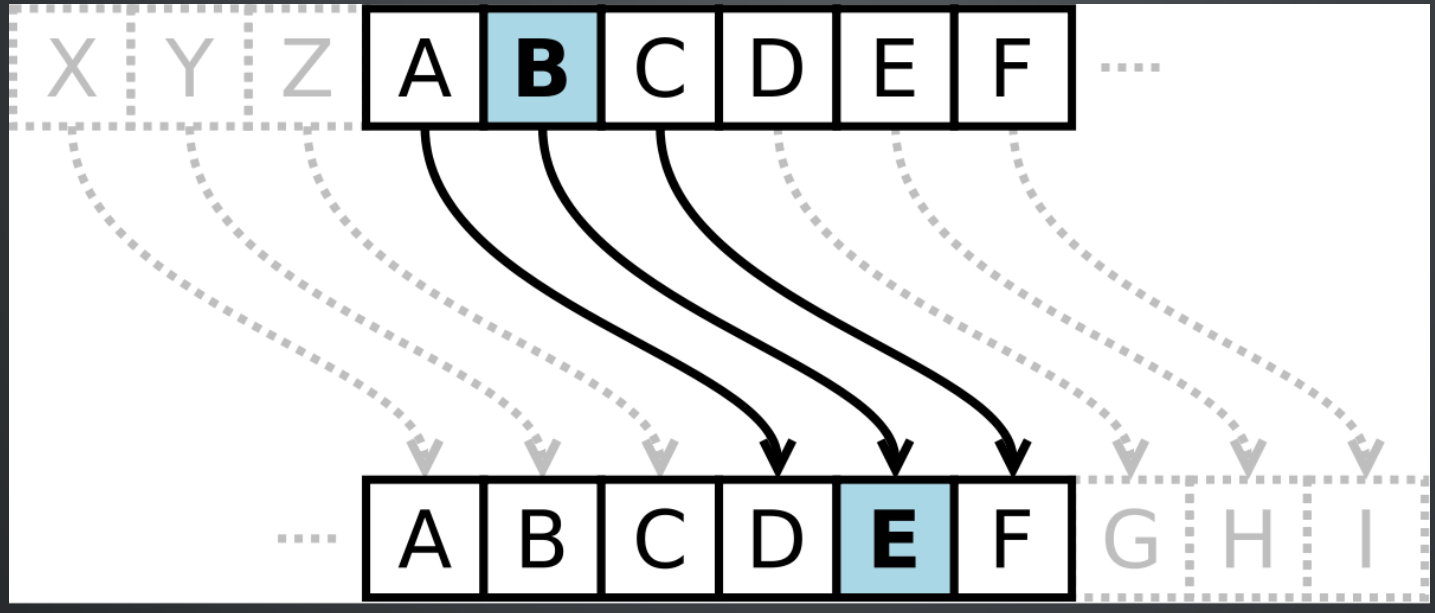
CRIPTOGRAFÍA

- Consiste en el estudio de técnicas que posibiliten transformar un mensaje legible en otro que no lo es. Sólo el emisor y el destinatario saben como devolverlo a su forma original, haciendo que cualquier otra persona que vea el mensaje cifrado no pueda entenderlo.
- Es parte de una disciplina antigua llamada "Criptología" que estudia las comunicaciones secretas.
- Usada entre otras finalidades para:
  - Autentificar la identidad de usuarios
  - Autentificar y proteger el sigilo de comunicaciones, transacciones, etc (confidencialidad)
  - Proteger la integridad de transferencias electrónicas de fondos

CRIPTOGRAFÍA CLÁSICA

Cifrados por sustitución, por transposición o combinaciones de ambas técnicas.

- Ejemplo por sustitución:
  - Caesar y ROT



CRIPTOGRAFÍA CLÁSICA

- Ejemplo de transposición columnar:
  - Scytale / Escítala

CRIPTOGRAFÍA CLÁSICA

- Ejemplo:
  - Máquina Enigma (2da guerra mundial)

SISTEMAS DE CRIPTOGRAFÍA MODERNA

Existen dos tipos básicos de criptosistemas:

- **Sistemas de cifrado simétrico** (también conocidos como sistemas de clave secreta o clave privada)
  - Ejemplos de algoritmos: 3DES, RC5, IDEA, AES, Blowfish
  - Dos modos de operación:
    - Cifrado en bloques
    - Cifrado de flujo
- **Sistemas de cifrado asimétrico** (también conocidos como sistemas de clave pública)
  - Ejemplos de algoritmos: Diffie-Hellman, RSA, DSA, ElGamal, CCE

EJEMPLO AES (ADVANCED ENCRYPTION STANDARD)

- Para poder leer un mensaje cifrado con un algorítmo criptográfico como AES (cifrado simétrico), se requiere conocer la "llave"/secreto/contraseña con la que fue cifrado. De otra manera, aunque conozcamos el procedimiento realizado sobre el mensaje, no podremos regresarlo a su forma original.
- Ejemplo "HOLA CLASE" encriptado con AES-128:

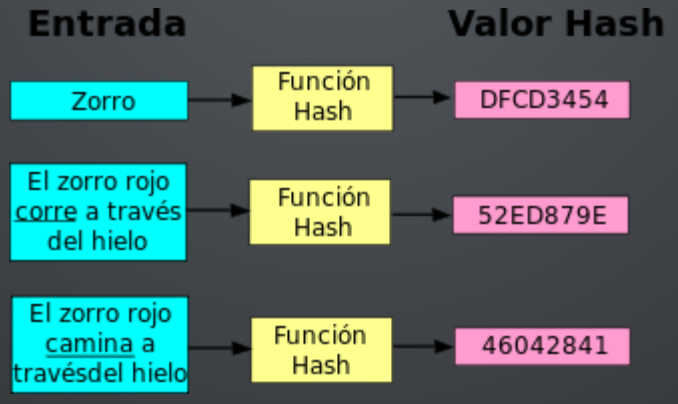
```
Clave secreta: "aesEncryptionKey"

Tamaño de la clave: 128 bits

Modo: ECB (Electronic Code Book)

Texto cifrado (hex): "e2bf2e420515f8143fd76ac64c075f47"
```

CODIFICACIÓN VS CRIPTOGRAFÍA

- Encoding: transformar los datos en otro formato usando un esquema disponible públicamente y puede ser fácilmente retransformado. Sirve para mantener la **usabilidad** de los datos. ***No puede ser visto como un mecanismo de seguridad.***
- Criptografía: transformar los datos en otro formato de cierta manera que solo determinadas personas puedan volver a ver el mensaje como estaba antes. Sirve para mantener la **confidencialidad** de los datos.

HASHING

- Es el proceso utilizado para generar una identificación única o "fingerprint" digital de una entrada, como, por ejemplo, una imagen de disco.
- Cuando se utiliza una función de hash segura, dos entradas no pueden generar el mismo hash.



HASHING EN LA FORENSIA DIGITAL

- Gracias a sus propiedades, las funciones de hash son utilizadas para comprobar que una evidencia no ha sido alterada durante la investigacion.
- Si un bit de la imagen es alterado, su hash será distinto al original.
- DEMO con HexEditor + md5sum.
  - Revisar como un bit cambia el hash

FUNCIONES DE HASH

PROPIEDADES

- El resultado es fácil de calcular.
- Es imposible obtener el mensaje original a partir del hash

EJEMPLOS:

- MD5 (128 bists, RFC 1321)
- SHA-1 (160 bits, NIST FIPS 180-2)
- SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
- SHA-3 (SHA-224, SHA-256, SHA-384, SHA-512)

¿QUÉ SE REPITE EN ESTAS IMÁGENES?

COLISIÓN DE HASHES (MD5)

```
$md5sum ship.jpg
253dd04e87492e4fc3471de5e776bc3d  ship.jpg

$ md5sum plane.jpg
253dd04e87492e4fc3471de5e776bc3d  plane.jpg
```

https://natmchugh.blogspot.com/2015/02/create-your-own-md5-collisions.html

ESTEGANOGRAFÍA

- Es la disciplina en la que se estudian y aplican técnicas que permiten el ocultamiento de objetos o mensajes, dentro de otros llamados portadores, de manera que no se perciba su existencia. Es una mezcla de artes y técnicas que se combinan para conformar la práctica de ocultar y enviar información sensible a través del portador, para que pueda pasar desapercibida.
- La esteganografía ha aparecido a lo largo de la historia de diferentes formas, cada cual más ingeniosa. Se conoce la existencia de mensajes ocultos en pequeñas bolitas de seda con mensajes escritos, envueltas en cera y usadas como adornos de los botones. Un escondite tan sutil para los mensajes secretos pasaba desapercibido para quienes intentaban interceptar a los mensajeros.

ESTEGANOGRAFÍA

- Otro método histórico de ocultar información se basaba en la grabación de mensajes en tablas de madera que después eran “borradas” con cera. De esta forma, las tablas parecían estar en un estado virgen, para poder ser escritas y sólo quien esperara recibir el mensaje derretía la cera para dejar la información oculta al descubierto.

ESTEGANOGRAFÍA

- La Esteganografía en el moderno sentido de la palabra y en terminos informáticos se refiere a información o a veces incluso a un archivo que se encuentra oculto dentro de otro. También permite implementar huellas digitales (copyright) o marcas de agua.
- Normalmente el contenedor es algún archivo del tipo **multimedia**, como una imagen digital, un video o un audio.

ESTEGANOGRAFÍA EN IMÁGENES ¿VES LAS 2 DIFERENCIAS?

ESTEGANOGRAFÍA EN IMÁGENES ¿Y AHORA?

DEMO STEGHIDE

- Embeber un archivo dentro de un JPG:

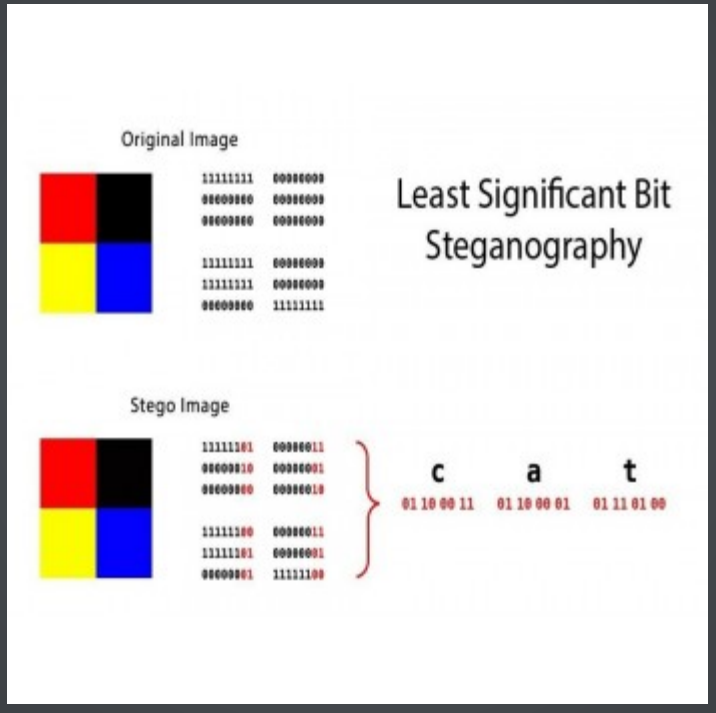
  steghide embed -cf [nombre_imagen] -ef [nombre_archivo_a_ocultar]

- Extraer el archivo oculto:

  steghide extract -sf [imagen_con_steganografia]

LEAST SIGNIFICANT BIT (LSB)

- Técnica de las más conocidas. Consiste en esconder un mensaje o archivo modificando los **bits menos significativos** de una imagen.

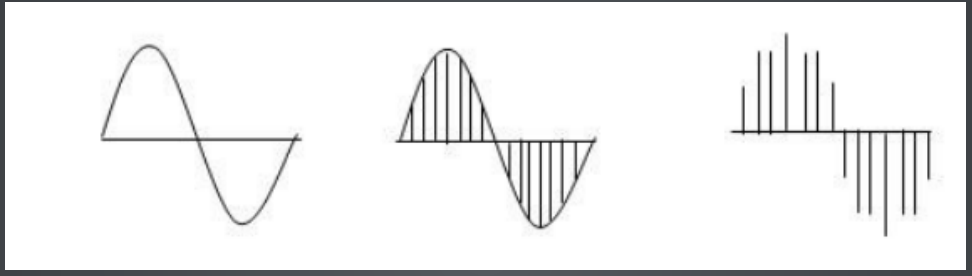


TIPOS DE ESTEGANOGRAFÍA

- **Pura**: se utiliza el estego-algoritmo para ocultar el mensaje o archivo esperando que quienes vean el archivo portador no detecten la modificación
- **De clave secreta**: se parametriza con una clave que define como aplicar el algoritmo

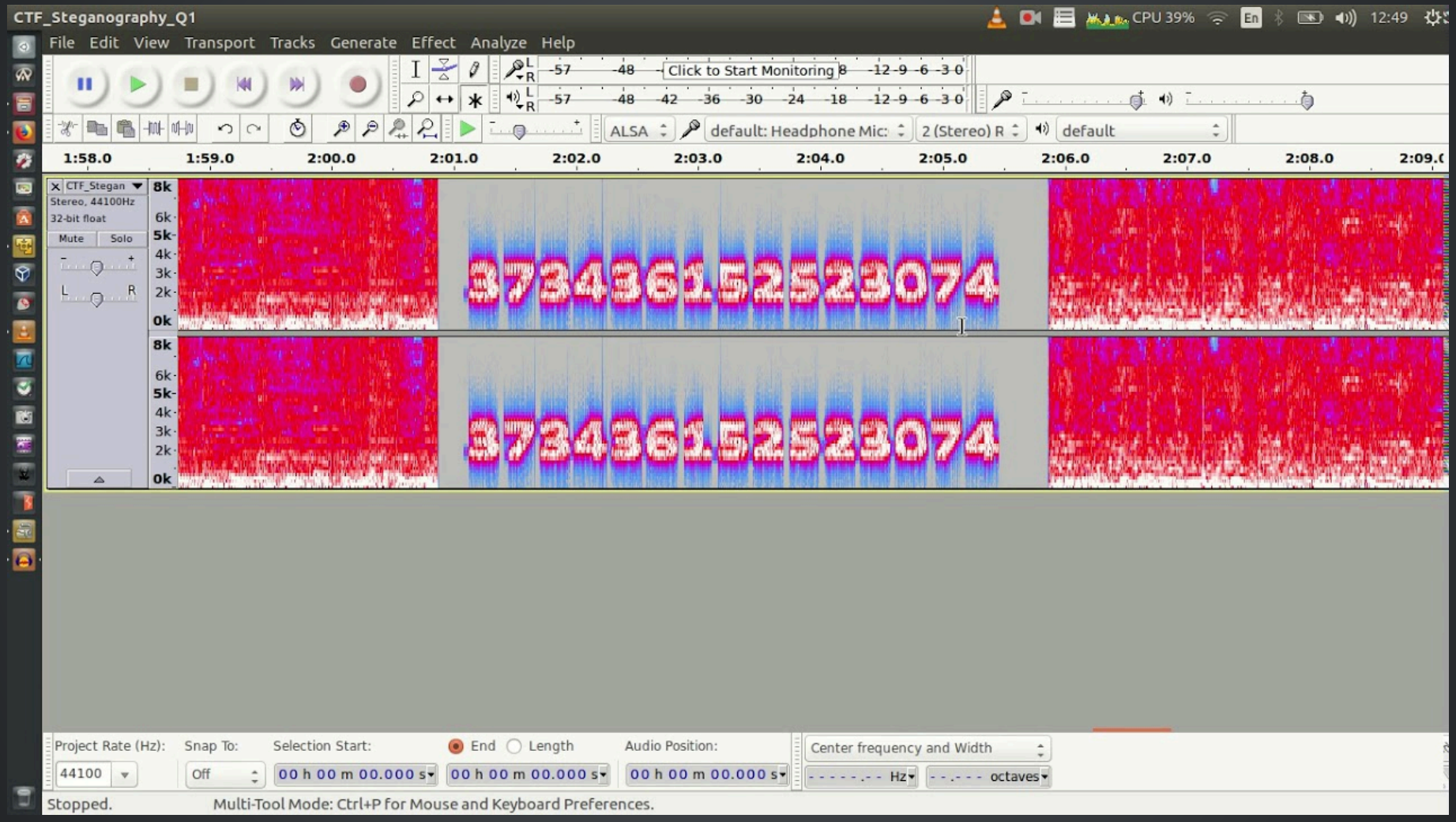
ESTEGANOGRAFÍA EN AUDIOS (LSB)

- Proceso de digitalización:

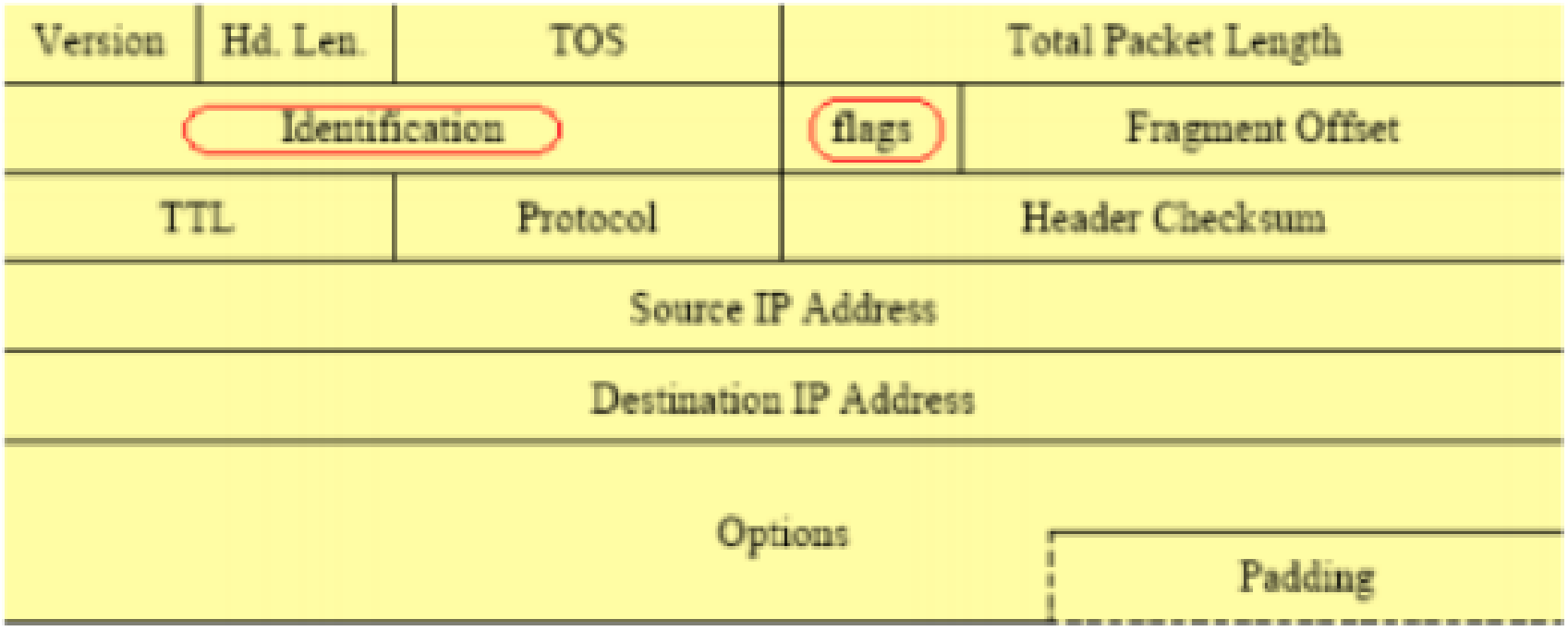


Flujo de audio	"A" en binario	Flujo de audio con el mensaje oculto
1001 1000 0011 1100	**0**	1001 1000 0011 110**0**
1101 1011 0011 1000	**1**	1101 1011 0011 100**1**
1011 1100 0011 1101	**1**	1011 1100 0011 110**1**
1011 1111 0011 1100	**0**	1011 1111 0011 110**0**
1011 1010 0111 1111	**0**	1011 1010 0111 111**0**
1111 1000 0011 1100	**1**	1111 1000 0011 110**1**
1101 1100 0111 1000	**0**	1101 1100 0111 100**0**
1000 1000 0001 1111	**1**	1000 1000 0001 111**1**

ESTEGANOGRAFÍA EN AUDIOS (ESPECTROGRAMA)



ESTEGANOGRAFÍA EN TRÁFICO DE RED



¿PREGUNTAS?